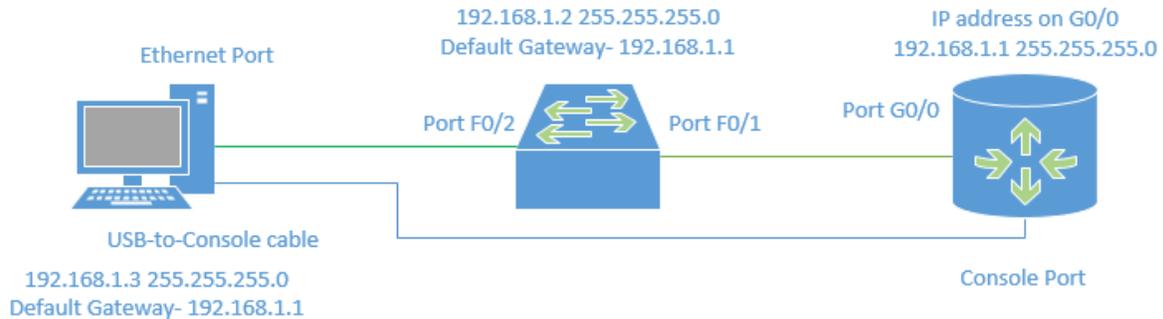
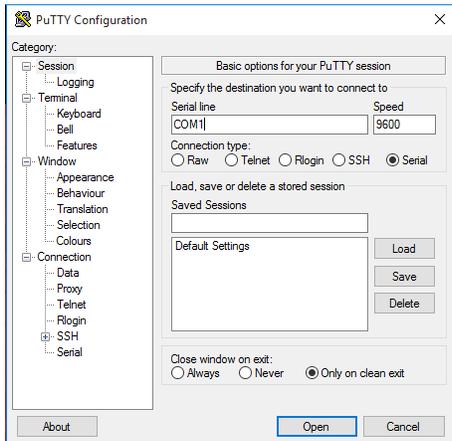


In this lab we will practice what is called a “basic configuration” on a Router. Regardless of the brand or model, the following configuration is standard procedure. We will be configuring a Cisco 1900 router.

Here is a diagram of how we will physically connect our lab:



- Step 1: Connect a rollover-to-RS232 cable (console cable) from your PC to the Router's Console port.
- Step 2: Connect a straight through cable from the switch's port F0/1 to the router's G0/0.
- Step 3: Start the Putty application on your PC.

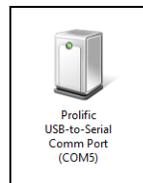


Choose “Serial”

In the serial line type the com port number your computer is assigning its serial port. The number will vary. (The software defaults to Com1)

To see what com number to use follow these steps:

1. In search type “devices and printers” and hit enter.
2. Look for the com port setting.
3. Type that number in putty and hit open



Step 4: Type in the following commands and settings. The info in red is a command, the info in black that follows changes a setting.



Let's get started: It is very important to pay close attention to your prompt. You must be in the correct prompt to issue commands. The prompt is the start of the command line (example: Router>)

The commands we will be using will be very similar to what you performed on a switch. There are a few differences. One big difference is port density. There are far less ports on a router to configure.

We will start with the console port. It must be secured.

Console Port Configuration:

```
Router>enable (This is a very important command, it moves you to "Privileged Exec" mode)
Router# configure terminal (This command takes you to "Global Configuration" mode)
Router(config)# line con 0 (This command tells the device we want to configure the "Line Console 0 Port")
Router(config-line)# logging synchronous (This is a strange command, when logging messages appear it resets the prompt)
Router(config-line)# exec-timeout 35 0 (The time the device waits before logging out when sensing no activity)
Router(config-line)# history size 200 (Sets history buffer size, or sets how many prior commands to remember)
Router(config-line)# password cisco1 (This is giving the word "cisco1" to the Line Console 0 port)
Router(config-line)# login (This is telling the device to make someone log in if accessing a device through the Line Con 0 port)
Router(config-line)# exit (This takes you back one step)
Router(config)#
```

To be able to connect to a device remotely it must have its VTY connections setup correctly. (VTY stands for *virtual teletype*)

Line VTY Port Configuration (if using username and passwords to access): This is the preferred method

```
Router(config)# line vty 0 4 (This command tells the device we want to configure "Line VTY ports 0 through 4")
Router(config-line)# logging synchronous (This is a strange command, when logging messages appear it resets the prompt)
Router(config-line)# exec-timeout 35 0 (The time the device waits before logging out when sensing no activity)
Router(config-line)# history size 200 (Sets history buffer size, or sets how many prior commands to remember)
Router(config-line)# login local (This tells the device to require a login and password if accessing the device through a VTY connection, and it searches a Local Database to find credentials)
Router(config-line)# exit (This takes you back one step)
Router(config)#
```

Line VTY Port Configuration (if using simple local password): Important!!! We are not going to perform this section

```
Router(config)# line vty 0 4
Router(config-line)# logging synchronous
Router(config-line)# exec-timeout 35 0
Router(config-line)# history size 200
Router(config-line)# password cisco1
Router(config-line)# login
Router(config-line)# exit
Router(config)#
```

Because the VTY lines can be accessed through a network connection, we rarely use a shared password for authentication. Best practice is to use individual login credential, which are stored in the devices Local Database.

The next series of commands will create a username and password in what is called the "Local Database"

We will also secure the enable command. It was the first command you issued when we started. The enable command is the gateway into changing configuration on a Cisco device. Since they are about 85% market share it is important to know how to secure it.

Security:

```
Router(config)# username admin password cisco2 (This creates a username and password entry in the devices local database)
Router(config)# enable password garbage (This command sets a password to the enable command. The enable command takes you into Privilege Exec mode)
Router(config)# enable secret cisco3 (This command sets an encrypted password to the enable command)
```



Next, let's do a little housekeeping. We will give the device a unique name, for Cisco it is called a host name. Next we will set a command to keep the device from searching the internet for incorrect commands. We will set a domain name, for Cisco devices this is important when using SSH to connect to a device.

Other

```
Router(config)# hostname OCARouter
OCARouter(config)# no ip domain-lookup
OCARouter(config)# ip domain name anyname.com
```

The next series of commands are important if we are going to securely connect to a device. Telnet can connect to a remote device, but because the information sent is not encrypted, it is best practice to not use it over an insecure connection. SSH is the preferred way to connect. To SSH we must create encryption keys. The following commands do this.

For SSH Key Generation:

```
OCARouter(config)# crypto key gen rsa (hit enter after rsa and specify the 1024 bit width)
OCARouter(config)# ip ssh ver 2
```

(These two commands set up SSH keys. They are used to encrypt data when using SSH to connect)

Now let's have some fun with the ports.

The Cisco 1900's in the classroom have 2 Ethernet Ports G0/0 and G0/1. They also have two serial ports. S0/0 and S0/1. The serial ports will be discussed later.

It is important to always know which ports you are configuring.

With Cisco routers we only work on one port at a time. Because routers can do so much more than a switch it is important to focus on one setting at a time.

First, let's run a couple of commands to see how our ports are configured. The following commands are called "show" commands. They are used to see current configuration settings. *Notice the prompt*

Show Commands:

```
OCARouter# show running-config (Shows the configuration changes you've made running in RAM)
OCARouter# show ip interface brief (Shows the state of you interfaces. Very valuable command)
OCARouter# show ip route (This show command displays a routers "Routing Table")
OCARouter# show ip protocol (This Show command displays protocols running on a router)
```

All changes you have made will be in your running configuration. The state of the ports is seen using the *show ip interface brief* command. The most important database on a router is its routing table. The *show ip route* command shows us what networks the router knows.

Now let's make some changes.

Let's start with the G0/0 interface.

To give a routers interface an IP address (For ether an Ethernet Port or a Serial Port):

```
OCARouter(config)# interface G0/0 (This command tells the device we want to configure the "Interface G0/0")
OCARouter(config-if)# ip address 192.168.1.1 255.255.255.0 (This gives the chosen interface and IP address and mask)
OCARouter(config-if)# no shut (This command turns the interface on)
OCARouter(config-if)# description This port is default gateway to 192.168.1.0 network (example of description)
OCARouter(config-if)# duplex Auto (A routers Ethernet interfaces can be set to two duplex settings; Full and Half)
OCARouter(config-if)# speed Auto (A routers Ethernet interfaces can be sent to speeds of: 10Mbs, 100, 1G, and Auto)
```

Run the show commands to see the changes.

It is important to save your work to the devices configuration file. Here is the command to save



OCARouter # **copy** running-config startup-config

The shorter version:

OCARouter# run start

Practice all the commands over and over till you feel comfortable. The goal is to be able to run all these commands from memory.